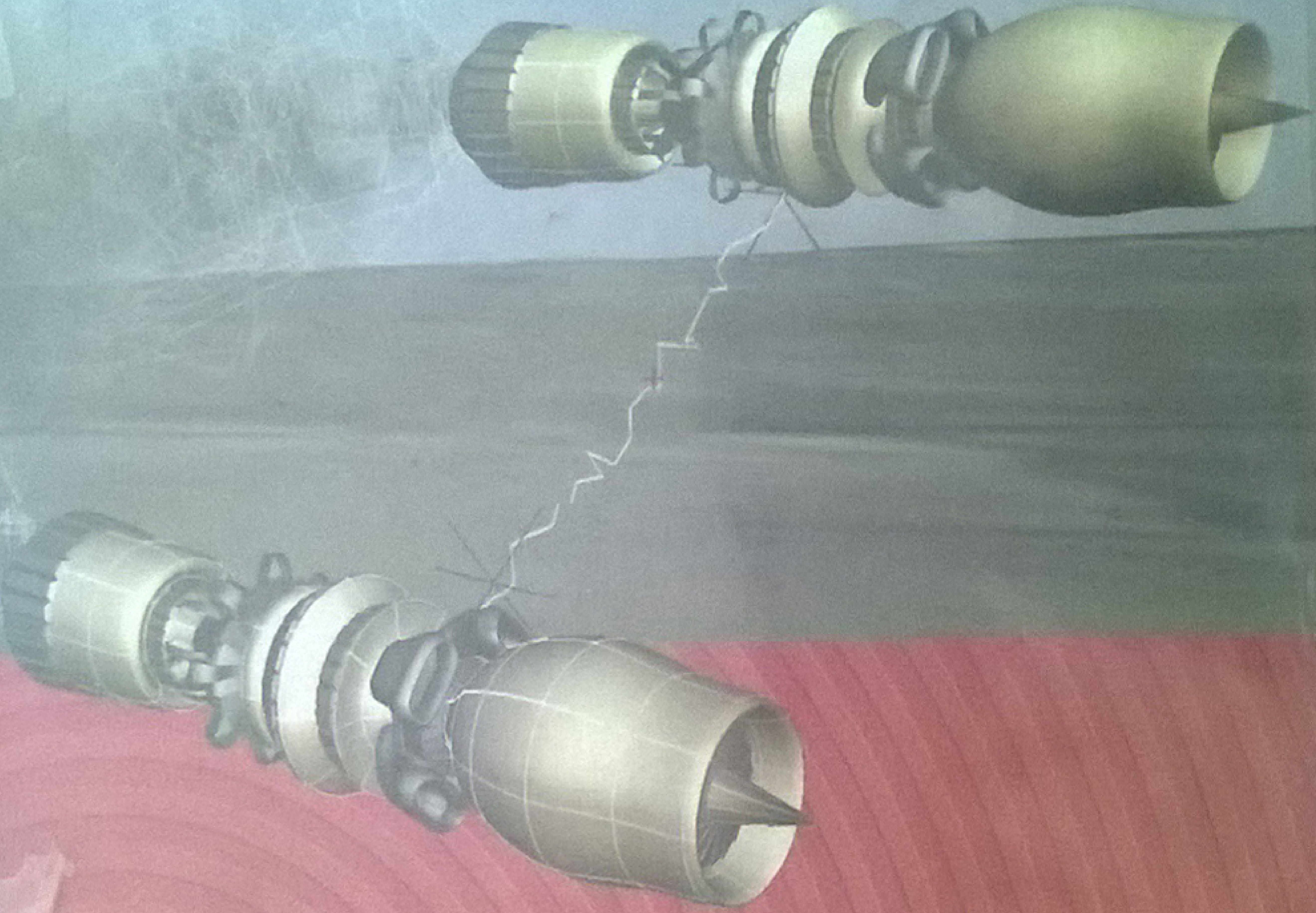
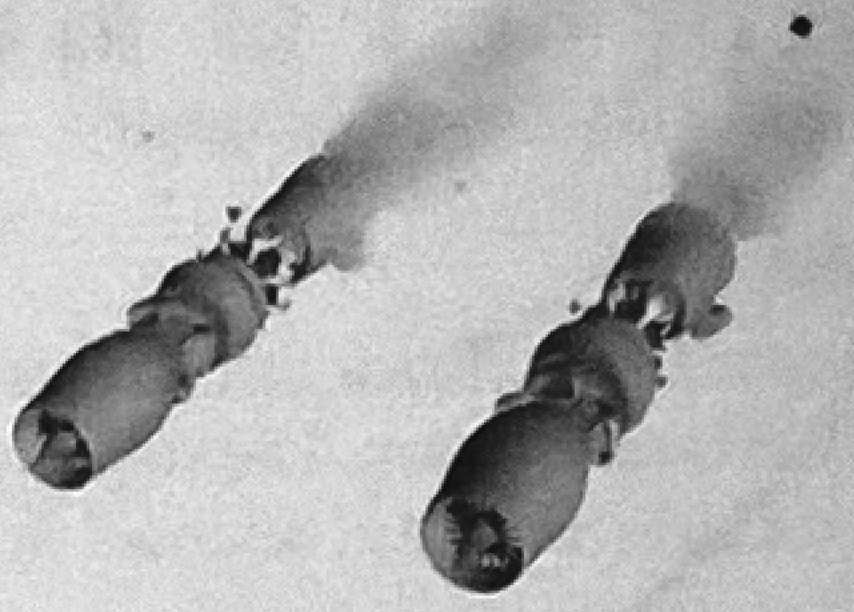
Learning Maya™ 5



Dynamics

Alias|wavefront®

Learning Maya™ 5



Dynamics

Alias|wavefront®

Learning Maya 5 | Dynamics

Credits:

John Patton, Lee Graft, Scyala Magloir

Special Thanks:

Brian Cho, Shawn Dunn, Erica Fyvie, Deion Green, Bob Gundu, David Haapalehto, Ken Ibrahim, Rachael Jackson, Danielle Lamothe, Robert MacGregor, Roland Reyer, Jason Schleifer, Carla Sharkey, Michael Stamler, Marcus Tateishi, Flavio Trevisan

© 2003 Alias | Wavefront, a division of Silicon Graphics Limited. All rights reserved.

This publication contains proprietary and confidential information of Alias | Wavefront, a division of Silicon Graphics Limited (the "Material"), and is protected by Federal copyright law and international intellectual property agreements and treaties. The Material may not be disclosed to third parties, translated, copied, or duplicated in any form, in whole or in part, or by any means, electronic, mechanical, photocopying, recording or otherwise, without the express prior written consent of Alias | Wavefront, a division of Silicon Graphics Limited ("A | W"). The Material is subject to change without notice. Neither A | W, nor its affiliates, nor their respective directors, officers, employees, or agents are responsible for punitive or multiple damages or lost profits or any other direct, indirect, special, indirect, incidental, or consequential damages including any damages resulting from loss of business arising out of or resulting from the use of the Material, or for technical or editorial omissions made therein.

THE MATERIAL IS PROVIDED "AS IS". A | W MAKES NO WARRANTIES, EXPRESS, IMPLIED OR ARISING BY CUSTOM OR TRADE USAGE RELATING TO THE MATERIAL, AND WITHOUT LIMITING THE GENERALITY OF THE FOREGOING, TO THE EXTENT PERMITTED BY APPLICABLE LAW, SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. NEITHER A | W NOR ITS AFFILIATES, PARENT COMPANIES, LICENSORS OR SUPPLIERS SHALL HAVE ANY LIABILITY WHATSOEVER IN CONTRACT, TORT OR OTHERWISE ARISING OUT OF OR IN CONNECTION WITH THE MATERIAL OR YOUR USE THEREOF

The Alias | Wavefront logo is a registered trademark and Alias | Wavefront, the Alias | Wavefront Education logo, Learning Maya, MEL, the Maya logo, and Maya Artisan are trademarks of Alias | Wavefront, a division of Silicon Graphics Limited. Maya is a registered trademark of Silicon Graphics, Inc., used exclusively by Alias | Wavefront, a division of Silicon Graphics Limited. All other trade names, trademarks and/or service marks are trademarks or registered trademarks of their respective owners.

Printed in Canada

ISBN 1-894893-42-5

10 9 8 7 6 5 4 3 2 1

TM
ALIAS | WAVEFRONT
EDUCATION

ALIAS | WAVEFRONT - 210 KING STREET EAST - TORONTO, ONTARIO - M5A 1J7

Maya Dynamics

Dynamically derived animation is a large subject. This book introduces and explores the various segments that encompass this corner of the Maya® universe.

WHAT IS MAYA DYNAMICS?

Dynamics in Maya uses rules of physics to let you simulate natural forces in your animation. Effects with complex motion such as smoke, rain, fire, or colliding objects lend themselves well to dynamically controlled animation. This animation is typically achieved by creating elements in a scene that react to the forces applied to them. By creating an environment of fields, expressions, goals, etc., the animator has artistic control over the affected objects, balancing the need for realism and animation requirements. Maya Dynamics is the animation of rigid bodies, soft bodies, and particles, the use of dynamic constraints, and the rendering strategies for hardware and software particle types.

After successfully completing the exercises in this book you will:

- Understand how dynamics in Maya work so you can create your own effects.
- Know how to work with rigid bodies and dynamic constraints.
- Be able to control particles with fields, ramps, and expressions.
- Understand hardware and software rendering techniques.
- Learn how to use particles to control the position and motion of instanced geometry.
- Be able to optimize and troubleshoot scenes involving dynamics.
- Know how to dynamically animate NURBS and polygonal surfaces using soft bodies.
- Understand how particle collisions work.

BOOK STRUCTURE

This book is project based. Each chapter contains several examples, usually with supporting data files where appropriate. In lessons where involved concepts are introduced, simple test beds are explored with and without example scene file data.

In many cases, the purpose of these lessons is to teach concepts and provide examples that demonstrate how something works while still providing techniques that readers can embellish for themselves in their own projects. The book is organized for a very hands-on progressive approach where the reader is strongly encouraged to work through the provided example data files using Maya.

Data files

Since there are data files for each chapter, we have created one subdirectory for each chapter containing all the files relevant to that chapter. These subdirectories are in the scenes folder of the Dynamics project. For example, the scenes for chapter one are in the *rigid_intro* directory, for chapter two the *rigid_constraints* directory, and so on. All of this data should be loaded from the CD that accompanies this book prior to proceeding with the exercises. It is recommended to follow the detailed instructions for installation that accompany the CD since preferences, movie files, and scripts are also provided as part of the data. It is recommended that you view the movies prior to doing each lesson so you have an idea of what you are trying to achieve in the lesson.

Who is this book designed for?

This book is intended for anyone who wants to learn more about using dynamics in Maya. It is especially useful for the effects artist or technical director who has experience with Maya or another dynamics package. The researcher or scientist using Maya to visualize or model physical phenomena will also find value from this book beyond what can be obtained from the Maya documentation and self-experimentation. The goal of this material is to provide something useful for a broad range of users in a short amount of time.

Although knowledge of MEL™ (Maya Embedded Language) and basic expression creation is not required, it is recommended and will make some of the MEL-based lessons easier to follow.

The following sections provide a very brief overview of some of the key topics discussed in this book:

Rigid Bodies

The rigid body system in Maya provides animation of geometric objects in a dynamically controlled, collision-based system.

- **Active and Passive Rigid Bodies**—Active and passive rigid bodies are created to collide and react with one another in a realistic manner. Active objects typically fall, move, spin, and collide with passive objects.
- **Rigid Body Constraints**—Rigid body constraints allow dynamic objects to be constrained or constrain each other. Spring, hinge, pin, etc. are some of the constraint types that will be explored.

Particles

Particles are objects that have no size or volume. They are reference points which are displayed, selected, animated, and rendered differently than other objects in Maya.

- **Particle Object and Array attributes**—Like other nodes in Maya, particles can be thought of as an object with a collective transform. They also contain attributes that control the individual particles using array attributes. The individual particle behavior can be controlled with ramps, scripts, and expressions.
- **Fields**—Fields such as gravity, turbulence, air, and others are used to easily move particles around your scene without the use of MEL and expressions.

- **Particle Expressions**—Particle expressions are a powerful and almost limitless method of controlling particle parameters. Particle expressions share the MEL syntax and methodology. Functions such as linstep(), sin(), and rand() provide mathematical control over particle appearance and motion.
- **Particle Collisions**—Particle collision events provide a method for creating and killing particles when they collide with geometry. Particle collision event procedures can be used to trigger specific MEL-scripted commands at collision time.
- **Emit function**—The emit function allows the user to create and position particles based on information directly derived from MEL and expressions. It requires an ample amount of MEL knowledge as more complicated usage of the emit function can get MEL-intensive.

Clip Effects

Clip effects provide the user with powerful and flexible tools for creating common dynamic effects. These are typically MEL scripts and expressions that automate the setup of the effect for the user. They also provide an excellent set of MEL and expression examples.

- **Flow**—The flow clip effect allows the user to select a curve as a motion path for particles.
- **Fire**—This clip effect will light an object on fire for both hardware and software rendering.
- **Smoke**—The smoke clip effect makes use of hardware sprites and is a good example of how to setup hardware sprites.

Particle Instancing

With particle instancing you can use particles to control the position and motion of instanced geometry

- **Animated Instance**—Particle instancing is only part of the functionality. You can instance a keyframed object to particles in the scene
- **Cycled Instance**—With the particle instancer you can cycle through a sequence of snapshot objects to create the instanced motion.
- **Software Sprites**—The particle instancer also provides aim control of the instanced object. If this instanced object is a textured plane, it can be aimed at the camera creating a software renderable sprite method.

Goals

Goals are a very powerful method of animating particles. Goals are a destination point that a particle wants to achieve. Particles can have multiple goal objects and per particle attributes designed specifically for goal-based interaction.

- **Goal weight and smoothness**—Goal weight and smoothness can be animated to provide particle movement that would be otherwise difficult to create with fields or expressions.
- **Per Particle Goal Attributes**—Goal attributes such as parentU and goalPP provide individual particle control. With the use of the parentId attribute, these values can be transferred from one particle to another.

Particle Rendering

What good is all this particle animation if you cannot render it out to contribute to the final shot?

- **Hardware Rendering**—Hardware rendering of particles provides a quick method of image creation. Typically, these images are then taken to the compositor who sweetens and integrates them with the rest of the scene elements.
- **Software Rendering**—Software rendering allows for scene integration of particles and rendered objects. Volumetric particle rendering is also created with software rendered particle types. Shadowing, glows, and other lighting effects are also combined.
- **Compositing**—Without compositing, much of this process would not be possible. Dynamics should always be viewed as another contributor to the elements that will make up the final image.

Softbodies

Animating geometry for fluid-like motion or with dynamic response is accomplished with softbody particles.

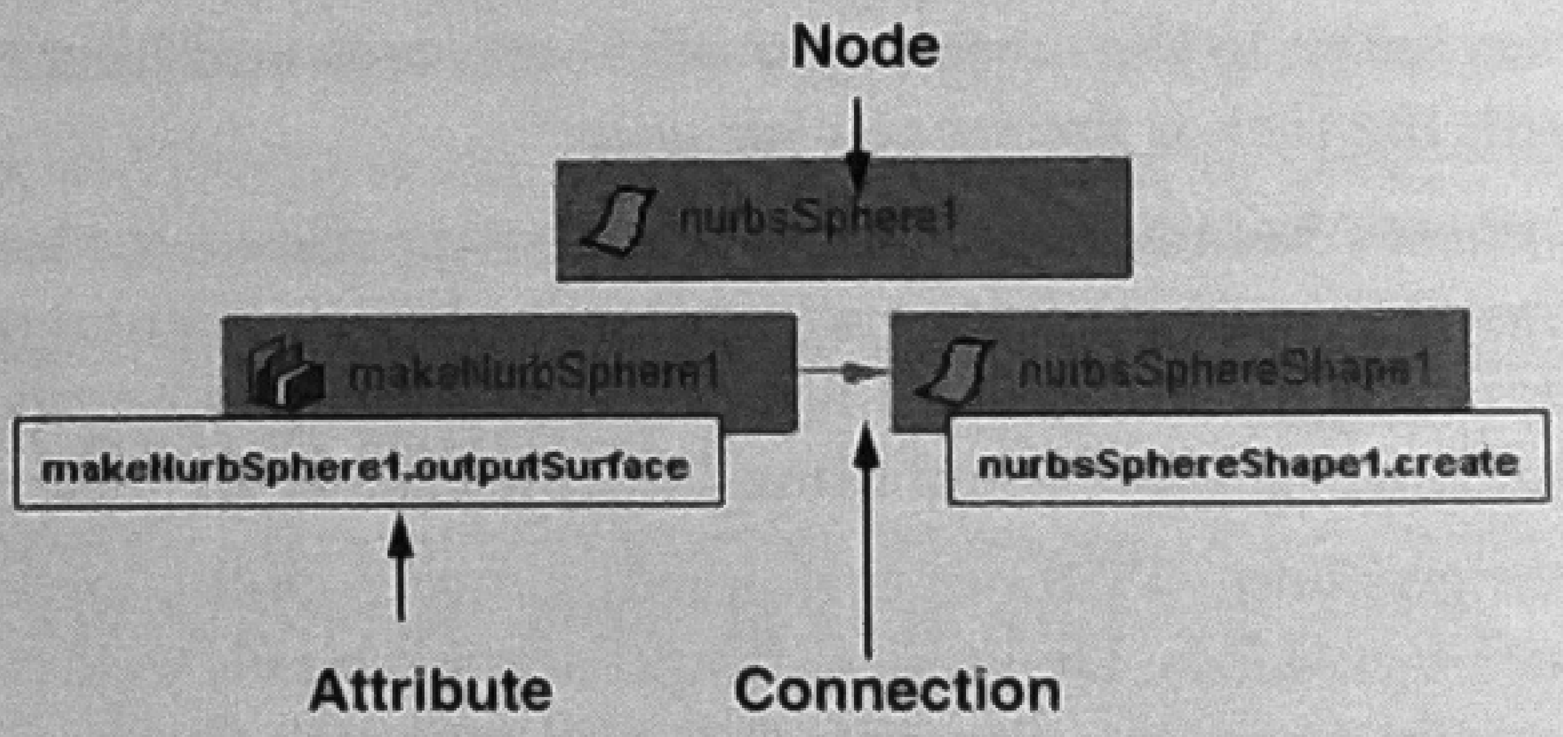
- **Softbodies**—Softbodies are geometry that have particles controlling the position and movement of the CVs or poly vertices of the objects.
- **Goals**—Goals are a fundamental part of controlling soft bodies. The goal weight controls the deviation of the soft body from its goal object.
- **Springs**—Although springs can be applied to any particle, even to dynamic geometry, they are especially suited to binding soft

body components together. Spring parameters such as stiffness, rest length, and damping can be controlled on a per object or per spring basis in Maya.

THE DEPENDENCY GRAPH

While creating dynamics, it is a good idea to have a basic knowledge of how Maya's system architecture works. Maya's system architecture uses a procedural paradigm that lets you integrate traditional keyframe animation, inverse kinematics, dynamics, and scripting on top of a node-based architecture that is called the **Dependency graph**. If you wanted to reduce this graph to its bare essentials, you could describe it as *nodes with attributes that are connected*. This node-based architecture gives Maya its flexible procedural qualities.

Below is a diagram showing a primitive sphere's Dependency graph as shown in the Hypergraph view. A procedural input node defines the shape of the sphere by connecting attributes on each node. This system is much like a production line. You can think of each node as contributing or performing some operation on the data that is passing through it.

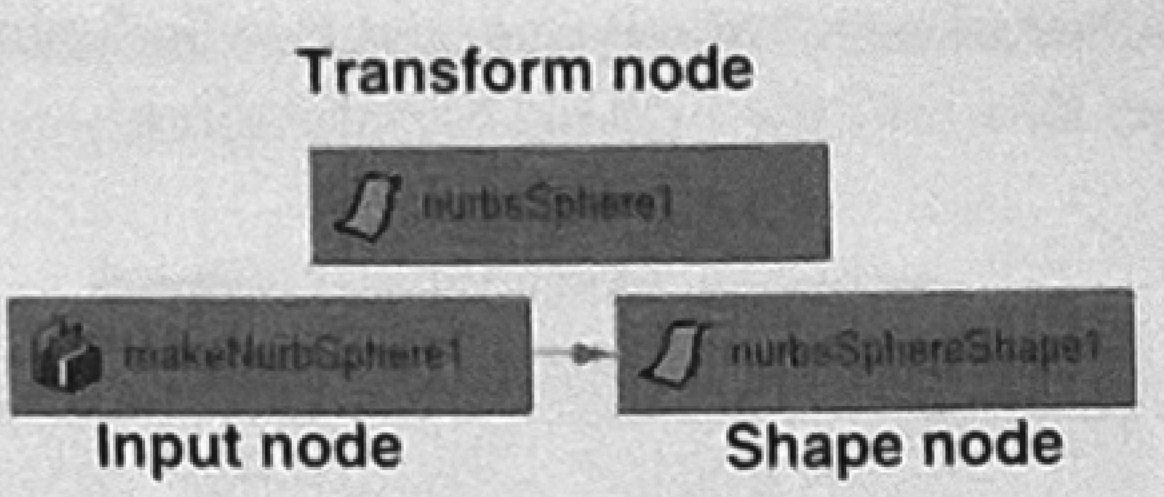


The Dependency graph

Nodes

Every element in Maya, whether it is a curve, surface, deformer, light, texture, expression, modeling operation, or animation curve, is described by either a single node or a series of connected nodes.

A *node* is a generic object type in Maya. Different nodes are designed with specific attributes so that each node can accomplish a specific task. Nodes define all object types in Maya including geometry, shading, and lighting. Shown below are three typical node types as they appear on a primitive sphere:



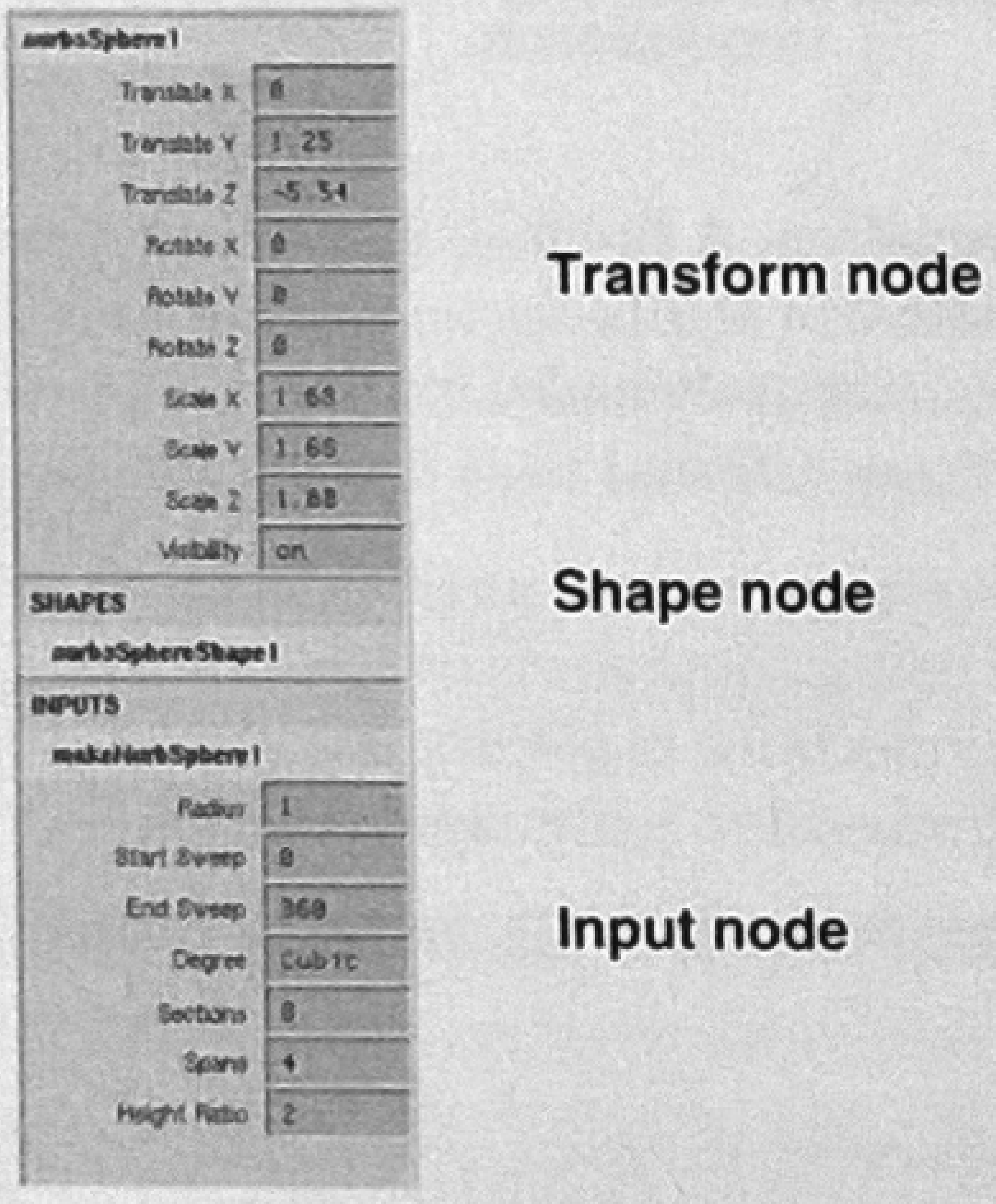
Node types on a sphere

Transform node—Transform nodes contain positioning information for your objects. When you move, rotate, or scale, this is the node you are affecting.

Input node—The input node represents options that drive the creation of your sphere's shape such as radius or endsweep.

Shape node—The shape node contains all the component information that represents the actual look of the sphere.

Maya's user interface presents these nodes to you in many ways. Below is an image of the Channel Box where you can edit and animate node attributes:



nurbsSphere1

Translate X	0
Translate Y	1.25
Translate Z	-5.54
Rotate X	0
Rotate Y	0
Rotate Z	0
Scale X	1.68
Scale Y	1.68
Scale Z	1.68
Visibility	on

SHAPES

nurbsSphereShape1

INPUTS

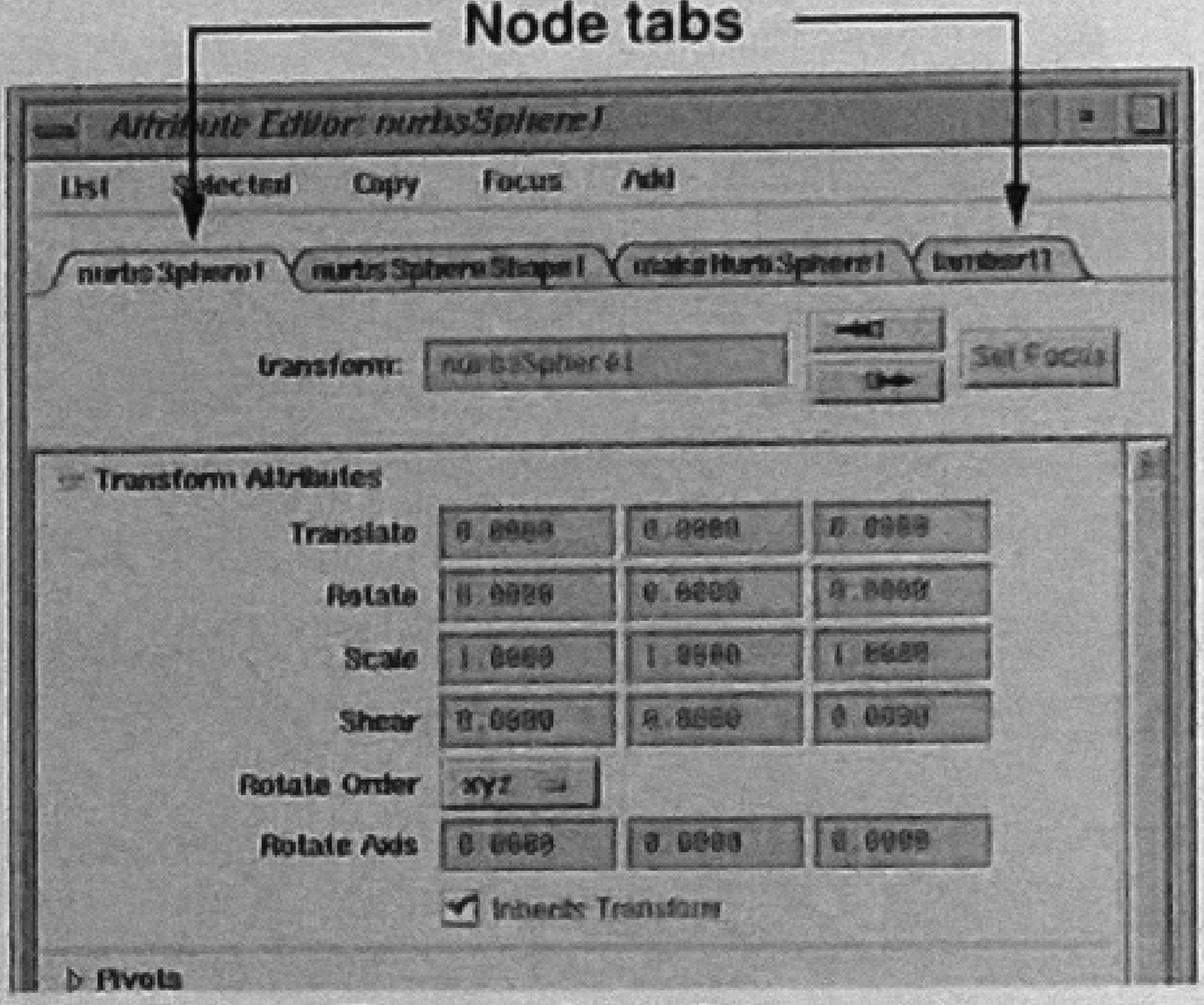
makeNurbSphere1

Radius	1
Start Sweep	0
End Sweep	360
Degree	Cubic
Sections	8
Spans	4
Height Ratio	2

Channel Box

Attributes

Each node is defined by a series of attributes that relate to what the node is designed to accomplish. In the case of a transform node, *X Translate* is an attribute. In the case of a shader node, *Color Red* is an attribute. It is possible for you to assign values to the attributes. You can work with attributes in a number of user interface windows including the Attribute Editor, the Channel Box and the Spread Sheet Editor.



transform: nurbsSphere1

Translate 0.0000 0.0000 0.0000
Rotate 0.0000 0.0000 0.0000
Scale 1.0000 1.0000 1.0000
Shear 0.0000 0.0000 0.0000
Rotate Order xyz
Rotate Axis 0.0000 0.0000 0.0000
☑ Inherits Transform

The Attribute Editor

One important feature in Maya is that you can animate virtually every attribute on any node. This helps give Maya its animation power. You should note that attributes are also referred to as *channels*.

Connections

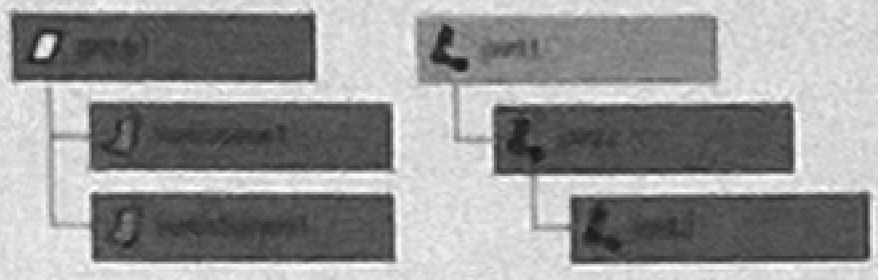
Nodes don't exist in isolation. A finished animation results when you begin making connections between attributes on different nodes. These connections are also known as *dependencies*. In the case of modeling, these connections are sometimes referred to as *construction history*.

Most of these connections are created automatically by the Maya user interface as a result of using commands or tools. If you desire, you can also build and edit these connections explicitly using the Connection Editor, by entering MEL (Maya Embedded Language) commands, or by writing MEL-based expressions.

Hierarchies

When you are building scenes in Maya, you can build dependency connections to link node attributes. When working with transform nodes or joint nodes, you can also build hierarchies which create a different kind of relationship between your objects.

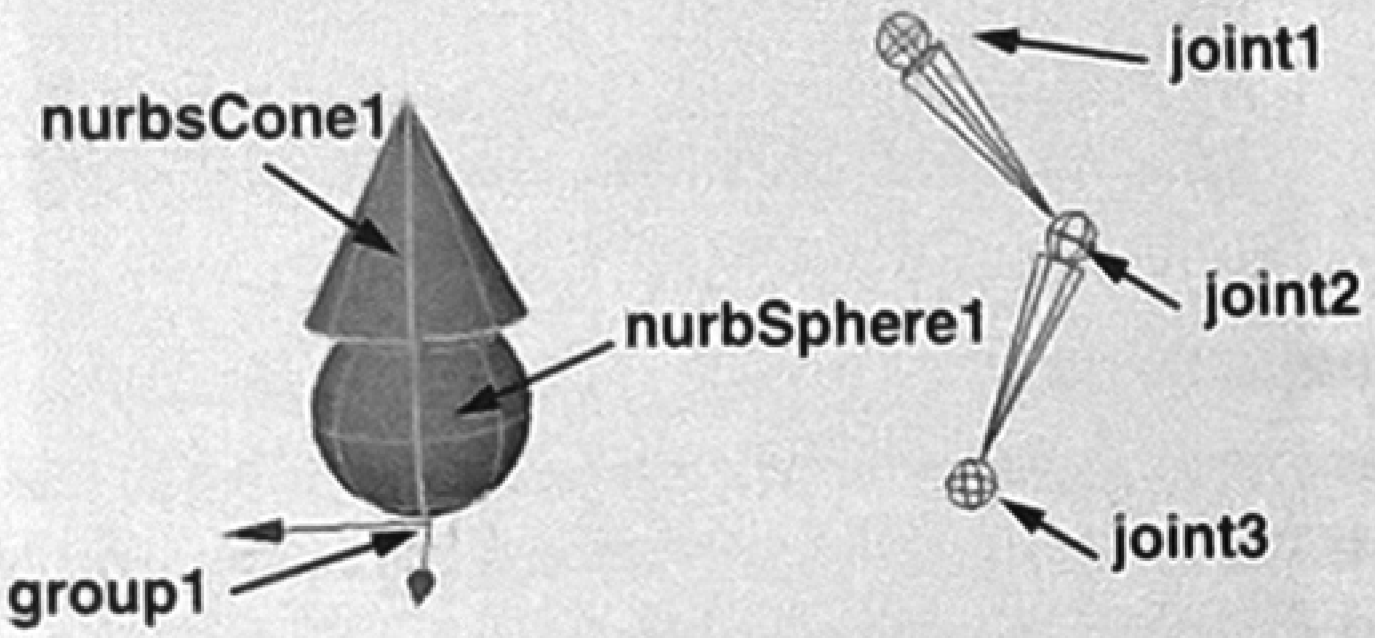
In a hierarchy, one transform node is *parented* to another. When Maya works with these nodes, Maya looks first at the top node, or *root* node, then down the hierarchy. Therefore, motion from the upper nodes is transferred down into the lower nodes. In the diagram below, if the *group1* node is rotated, then the two lower nodes will rotate with it. If the *nurbsCone* node is rotated, the upper nodes are not affected.



Object and joint hierarchy nodes

Joint hierarchies are used when you are building characters. When you create joints, the joint pivots act as limb joints while bones are drawn between them to help visualize the joint chain. By default, these hierarchies work just like object hierarchies. Rotating one node rotates all of the lower nodes at the same time.

When you are working with characters, you will use *inverse kinematics* to reverse the flow of the hierarchy.



Object and joint hierarchies

The Hypergraph

In Maya, you can visualize hierarchies and dependencies using the Hypergraph. The following steps demonstrate how to work with various node types in the Hypergraph:

Working with hierarchies and dependencies

If you understand the idea of *nodes with attributes that are connected*, then you will understand the Dependency graph. You can see what this means in Maya by building a simple primitive sphere.

1 Set up your view panels

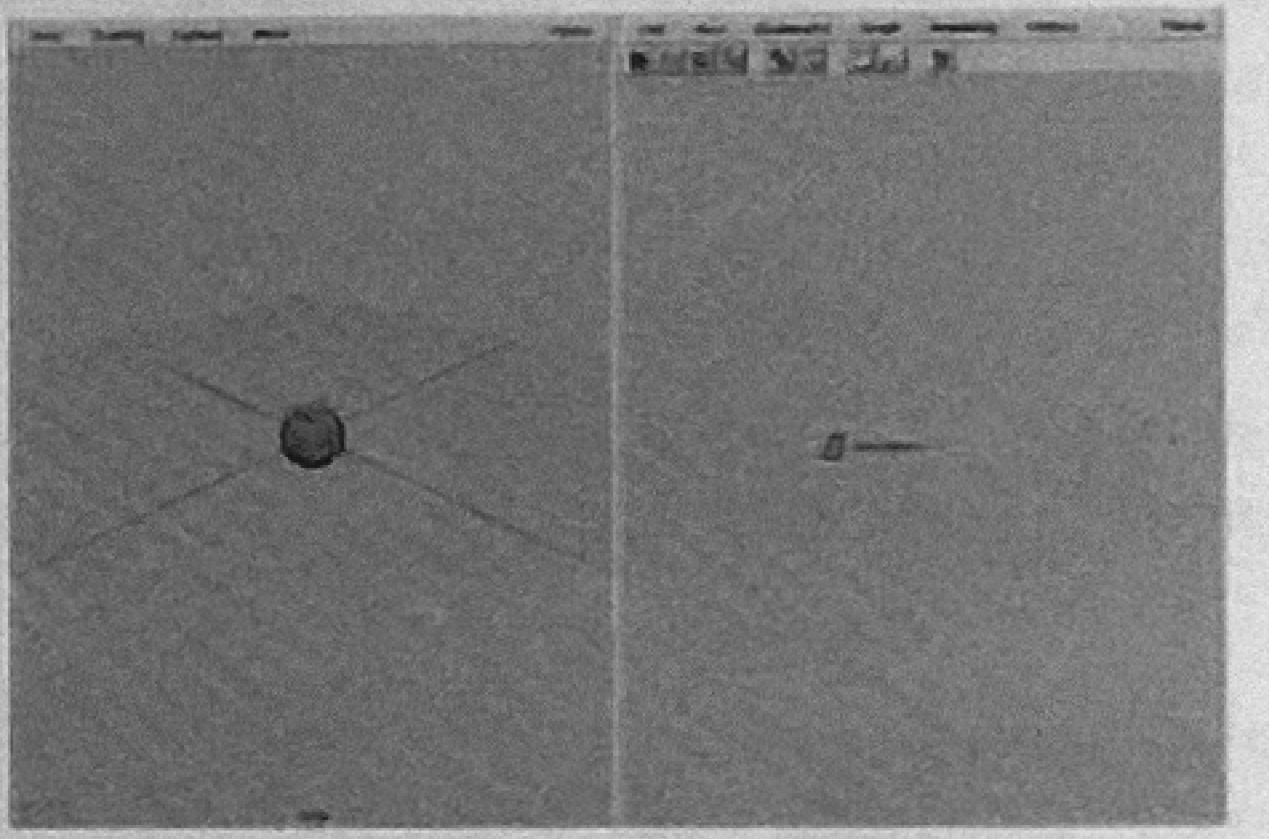
To view nodes and connections in a diagram format, the Hypergraph panel is required along with a Perspective view.

- Select **Panels → Layouts → Two Panes Side by Side**.
- Set up a Perspective view in the first panel and a Hypergraph view in the second panel.

- Dolly into the Perspective view to get closer to the grid.

2 Create a primitive sphere

- Go to the modeling menu set.
- Select **Create → NURBS Primitives → Sphere**.
- Press **5** to turn on smooth shading and **3** to increase the surface smoothness of the sphere.



New sphere

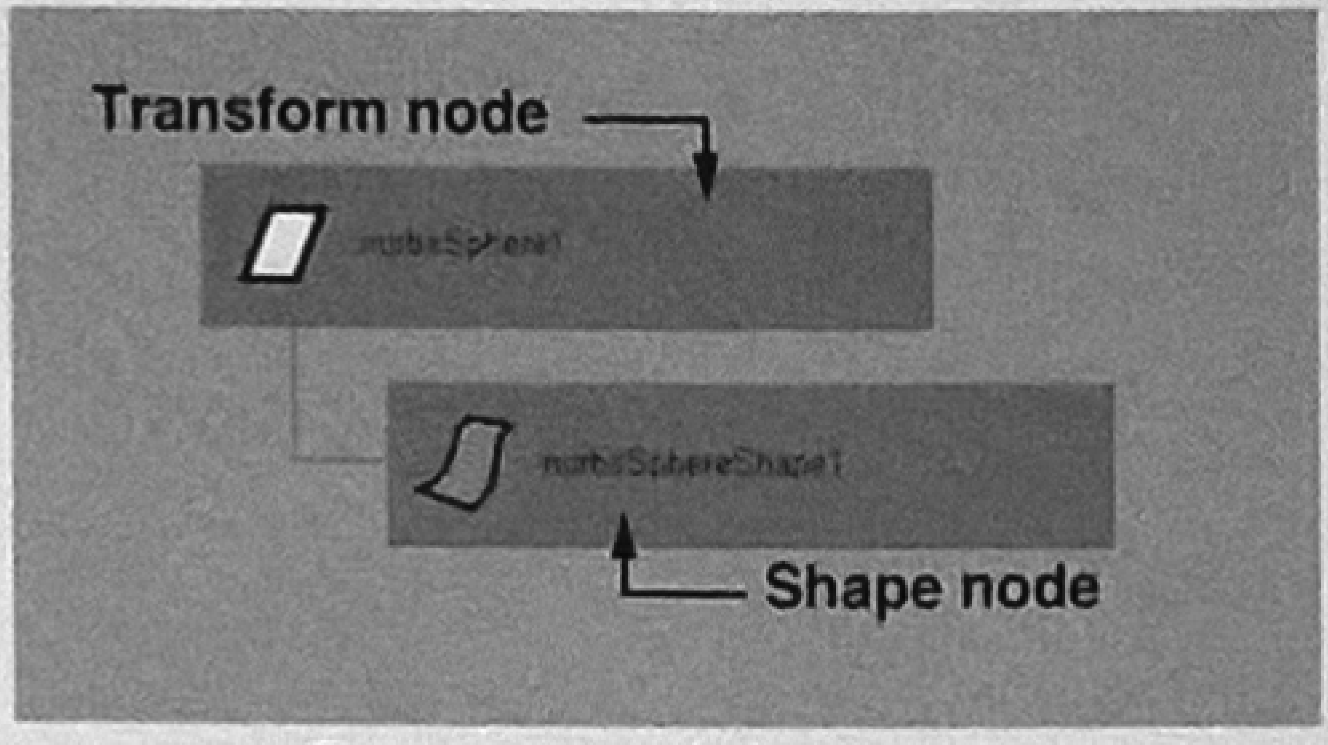
3 View the shape node

In the Hypergraph panel, you are currently looking at the scene view. The scene view is focused on *transform nodes*. This node type lets you set the position and orientation of your objects.

Right now, only a *nurbsSphere* node is visible. In actual fact, there are two nodes in this hierarchy but the second is hidden by default. At the bottom of most hierarchies, you will find a *shape node* which contains the information about the object itself.

- In the Hypergraph panel, select **Options → Display → Shape nodes**.

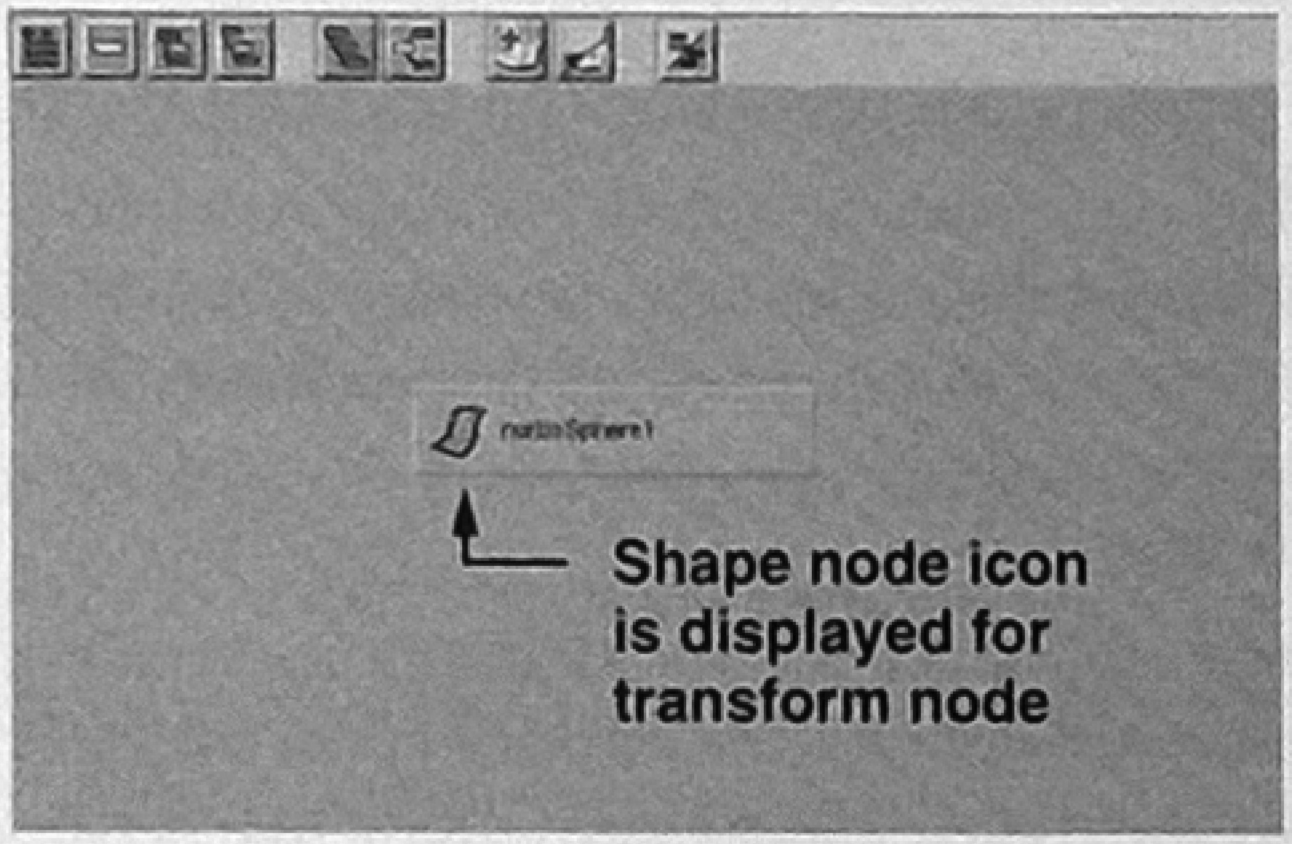
  You can now see the *transform node* which is, in effect, the positioning node and the *shape node* which contains information about the actual surface of the sphere. The transform node defines the position of the shape below it.


Transform node
nurbsSphere1
nurbsSphereShape1
Shape node

Transform and shape nodes

- In the Hypergraph panel, select **Options** → **Display** → **Shape nodes** to turn these off.

  You will notice that when these nodes are expanded, the shape node and the transform node have different icons. When collapsed, the transform node takes on the shape node's icon to help you understand what it going on underneath.

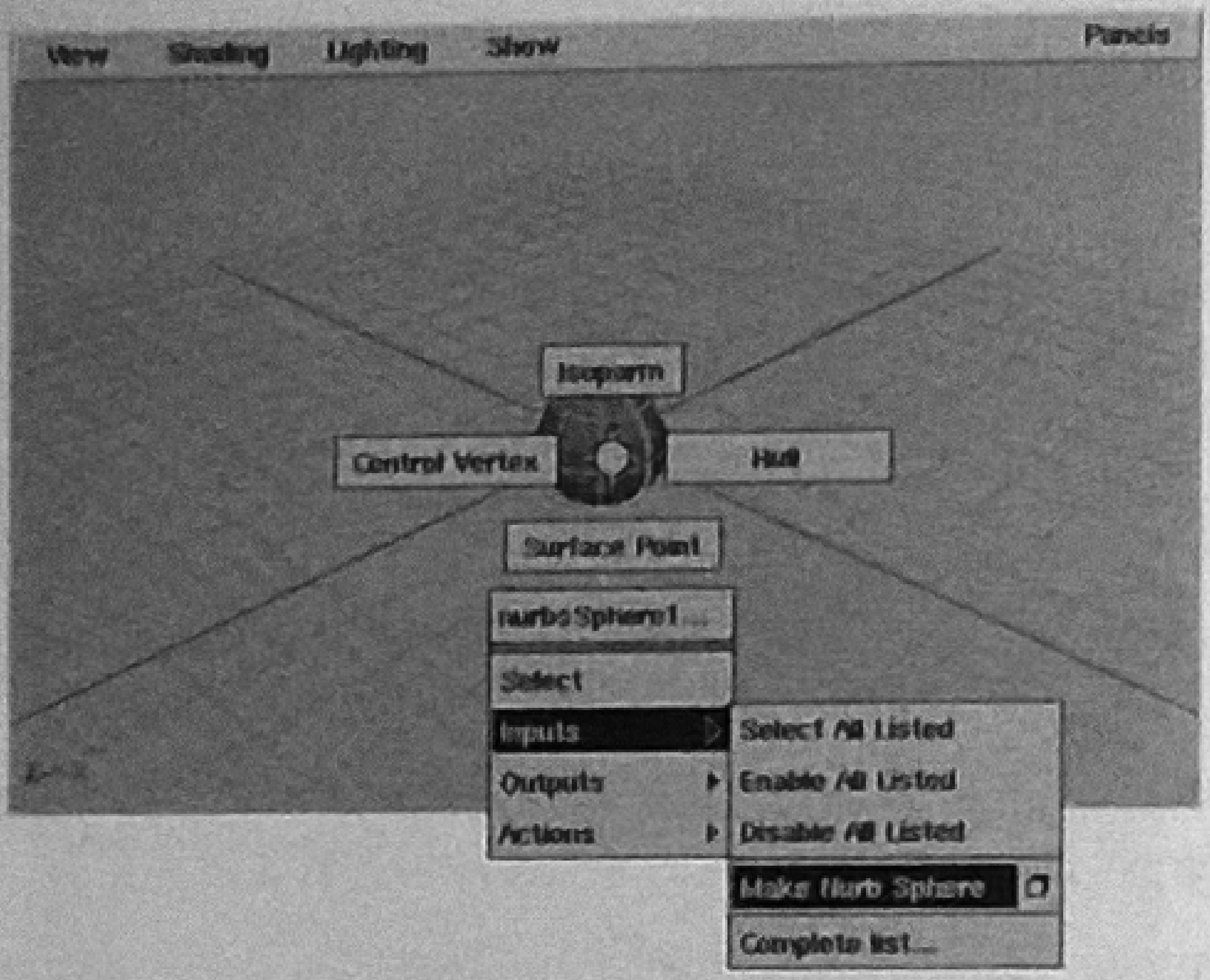

nurbsSphere1
Shape node icon is displayed for transform node

Transform node on its own

4 View the dependencies

To view the dependencies that exist with a primitive sphere, you need to take a look at the Input and Output connections.

- Click on the sphere with the right mouse button and select **Inputs** → **Make Nurb Sphere** from the marking menu.


View
Shading
Lighting
Show
Panels
Isoparm
Control Vertex
Hull
Surface Point
nurbsSphere1...
Select
Inputs
Outputs
Actions
Select All Listed
Enable All Listed
Disable All Listed
Make Nurb Sphere
Complete list...

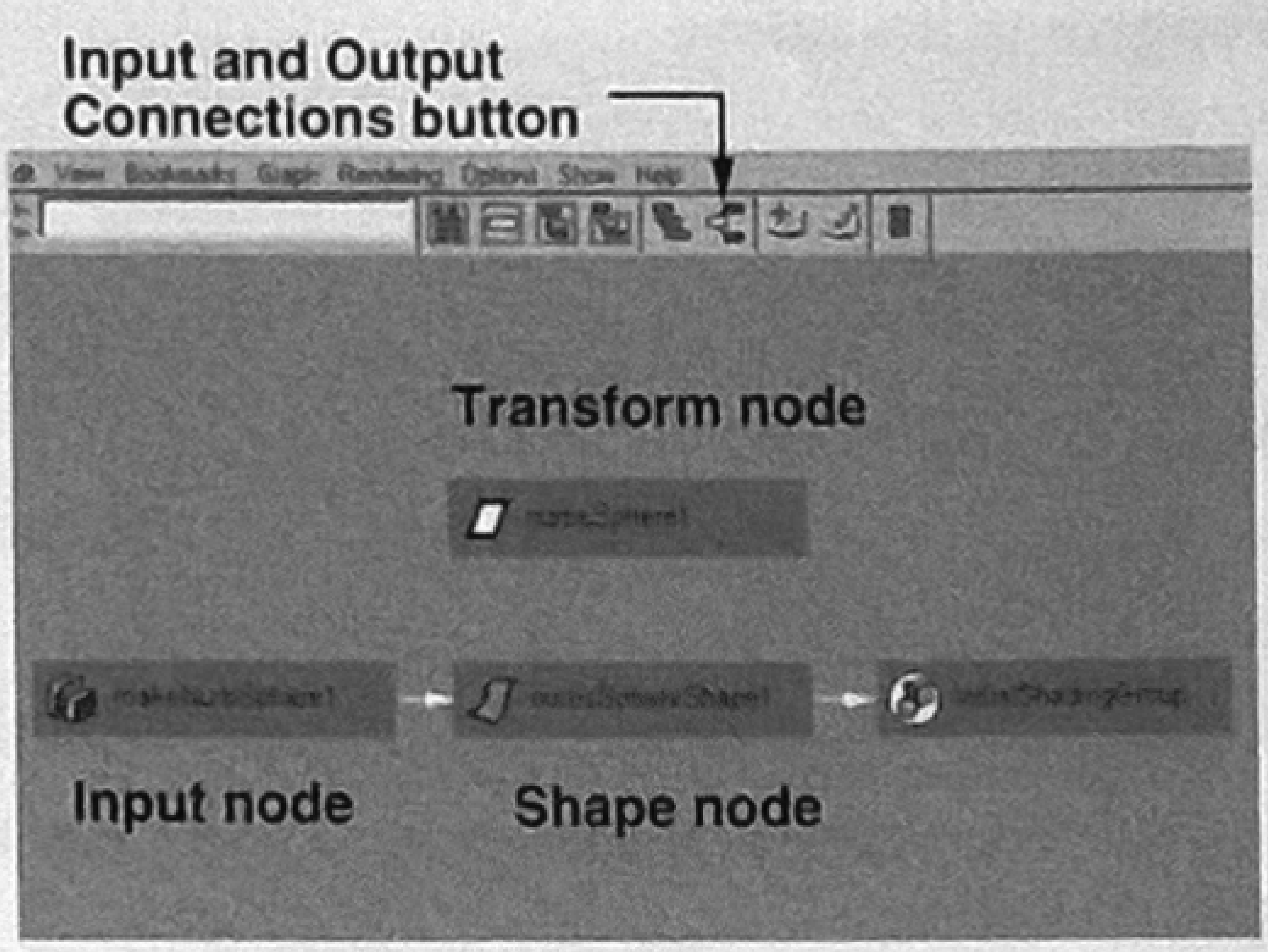
Selection marking menu

Note: You can also select the input node by choosing it in the Channel Box.

- In the Hypergraph panel, click on the **Input and Output Connections** button.

  See the original transform node which is now separated from the shape node. While the transform node has a hierarchical relationship to the shape node, their attributes are not dependent on each other.

  The *input node* called *makeNurbSphere* is a result of the original creation of the sphere. The options set in the Sphere Tool's option window have been placed into a node that feeds into the shape node. The shape node is dependent on the input node. If you change values in the input node, then the shape of the sphere changes.



Sphere dependencies

5 Edit the attributes in the Channel Box

In the Channel Box, you can edit attributes belonging to all of the node types. This lets you affect both hierarchical relationships and dependencies.

If you edit an attribute belonging to the *makeNurbsSphere* node, then the shape of the sphere will be affected. If you change an attribute belonging to the *nurbsSphere* transform node, then the positioning will be changed. Using the Channel Box will help you work with the nodes.

- For the transform node, change the **Rotate Y** value to **45**.
- For the input node, change the **Radius** to **3**.

  You can set attribute values to affect either the scene hierarchy or the Dependency graph.

Animating the sphere

When you animate in Maya, you are changing the value of an attribute over time. Using keys, you set these values at important points in time, then use tangent properties to determine how the attribute value changes between the keys.

The key and tangent information is placed in a separate animation curve node that is then connected to the animated attribute.

1 Select the sphere

- In the Hypergraph panel, click on the **Scene Hierarchy** button.
- **Select** the *nurbsSphere* transform node.

2 Return the sphere to the origin

Since you earlier moved the sphere along the three axes, it's a good time to set it back to the origin.

- In the Channel Box, change the **Rotate Y** attribute to **0**.

3 Animate the sphere's rotation

- In the Time slider, set the playback range to **120** frames.
- In the Time slider, go to frame **1**
- Click on the **Rotate Y** channel name in the Channel Box.
- Click with your right mouse button and select **Key Selected** from the pop-up menu.

  This sets a key at the chosen time.
- In the Time slider, go to frame **120**.
- In the Channel Box, change the **Rotate Y** attribute to **720**.
- Click with your right mouse button and select **Key selected** from the pop-up menu.
- Playback the results.

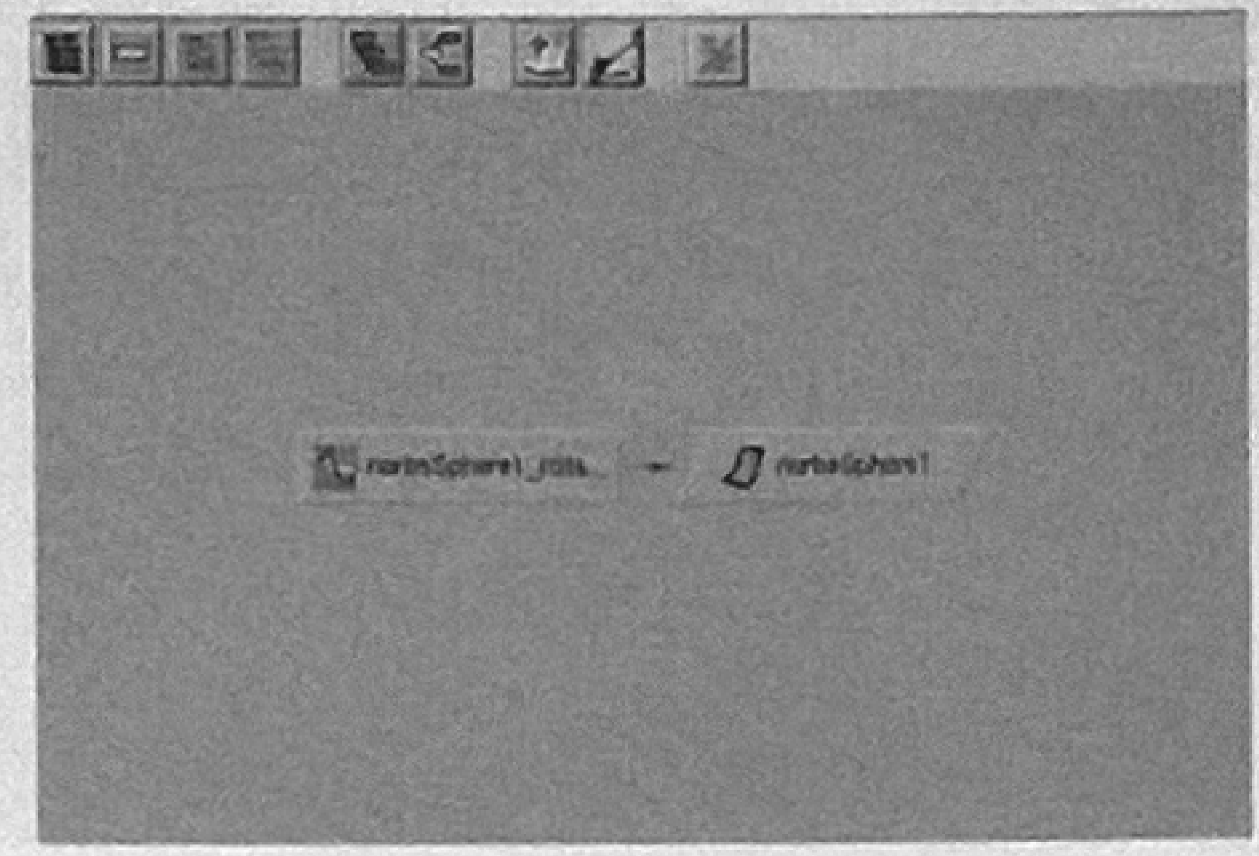
  The sphere is now spinning.

4 View the Hypergraph dependencies

- In the Hypergraph panel, click on the **Input and Output Connections** button.

  You see that an animation curve node has been created and connected to the transform node. The transform node is now shown as a trapezoid to indicate that it is connected to the animation curve node. If you click on the connection arrow, you will see that the connection is to *Rotate Y*.

  If you select the animation curve node and open the Attribute Editor, you will see that each key has been recorded along with value, time, and tangent information. You can actually edit this information here, or use the Graph Editor where you get more visual feedback.



Connected animation curve node

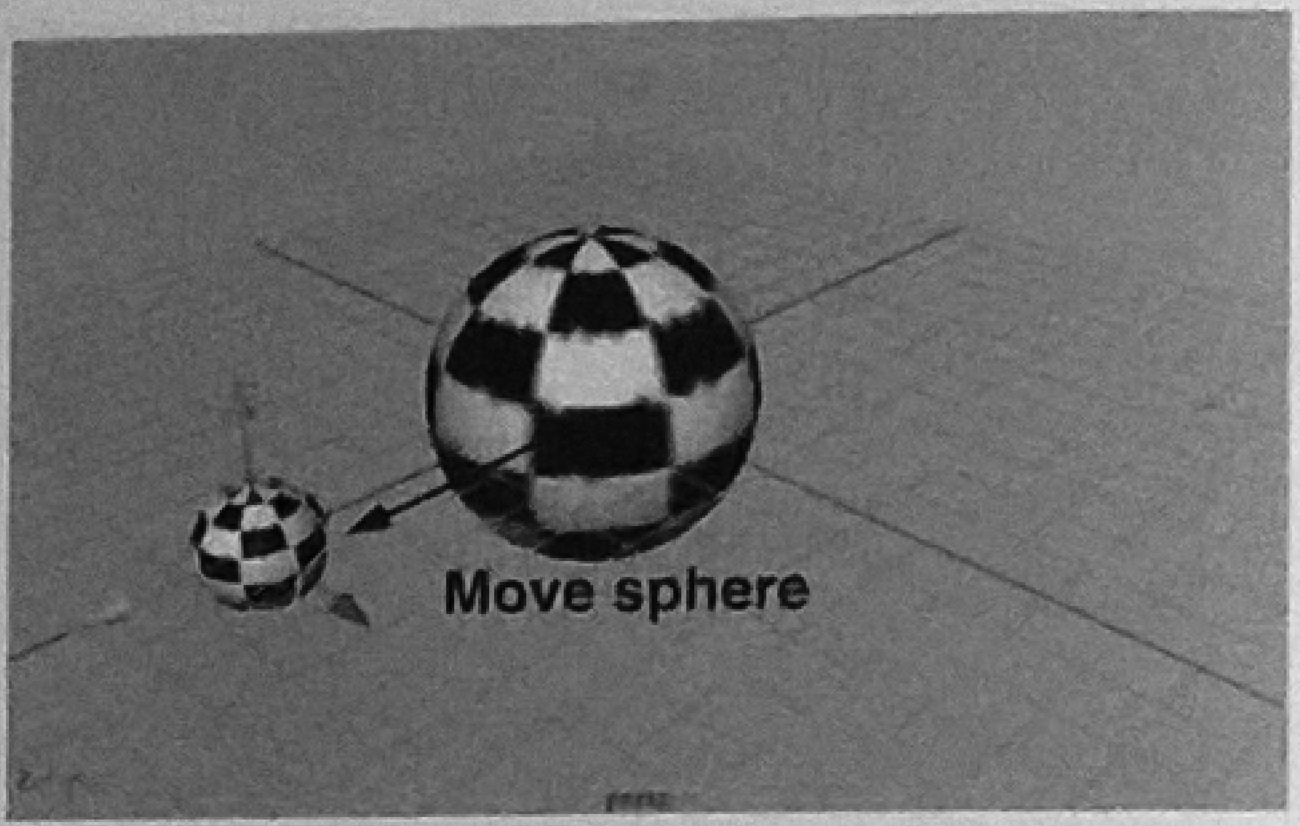
Parenting in the Hypergraph

So far, you have worked a lot with the dependency connections but not with the scene hierarchy. In a hierarchy, you always work with transform nodes. You can make one transform node the *parent* of another node, thereby creating a child which must follow the parent.

You will build a hierarchy of spheres that are rotating like planets around the sun. This example is a helpful way to understand how scene hierarchies work:

1 Create a new sphere

- In the Hypergraph panel, click on the **Scene Hierarchy** button.
- Go to the **Modeling** menu set.
- Select **Create → NURBS Primitives → Sphere**.
- **Move** the sphere along the Z axis until it sits in front of the first sphere.
- Press **3** to increase the display smoothness of the sphere.
- Go to the **Hypershade** menu.
- Apply a checker shader to both spheres.

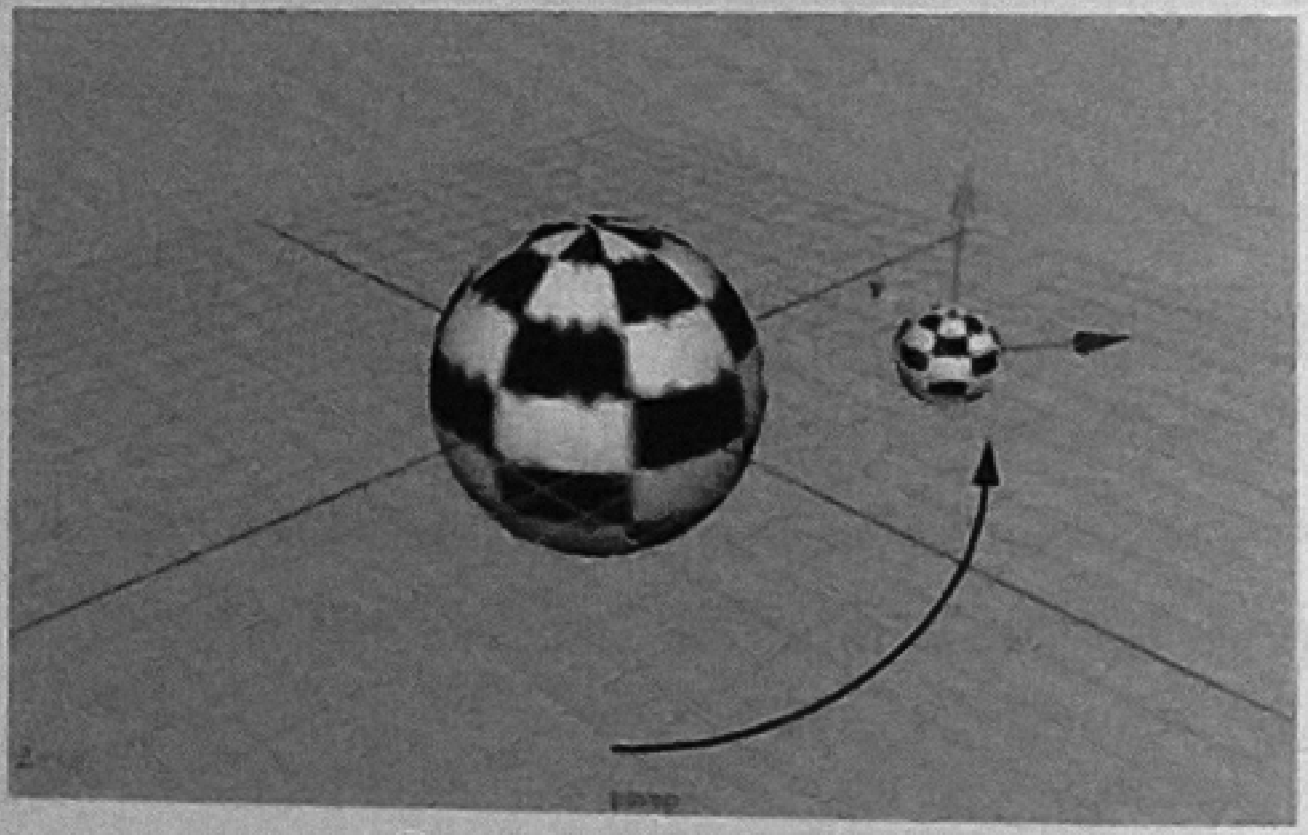


Second sphere

2 Parent the sphere to the first sphere

- In the Hypergraph, drag the node icon with **MMB** for the second sphere onto the first sphere. Now they are parented together.
- Playback the scene.

  The second sphere rotates along with the first sphere. It has inherited the motion of the original sphere.



Rotating hierarchy

SUMMARY

Venturing into Maya's dynamics can seem somewhat daunting when you look at all of the options. However, this is one of the most interesting and powerful aspects of the software. The material in this book is presented in a progressive manner to take you through an entire range of tools and techniques that are available using Maya's dynamics system.

This book, like many of the learning products from Alias | Wavefront™, encourages the importance of building a foundation of understanding of Maya in a broader sense. This will allow you to take advantage of Maya's entire tool set to solve the problems and design the effects you want to achieve. For example, understanding Maya's hierarchy and Dependency graph is essential for understanding the "how" and "why" of designing and troubleshooting the effects you create in Maya. The exercises in this book will provide a good starting place for you and show you what is possible. You should always feel free to experiment with your own techniques and take these presented concepts in a direction that makes the most sense for the work you are doing.

Want to Learn More?

Alias|Wavefront publishes a variety of self-study learning materials that can help you improve your skills.

the art of maya
Games and Interactive
Character Rigging and Animation
Maya

Visit

www.aliaswavefrontstore.com

and check out our books and training materials.

Featured Products:

Exploring Particles

Maya Techniques | Exploring Particles is your key to creating advanced particle effects. Take advantage of helpful data files so that you can begin to manipulate scenes and expressions. Enriched with real life production examples, this DVD will allow you to understand how to build and render particles to achieve the results you want.

Understanding Fluid Effects

Maya Techniques | Understanding Fluid Effects features Duncan Brinsmead and Rick Kogucki to give you an inside look into Maya Fluid Effects. Learn to simulate and render fluid motion, and create a range of effects from gaseous atmospherics, such as clouds and smoke, to thick viscous liquids, like lava and mud.

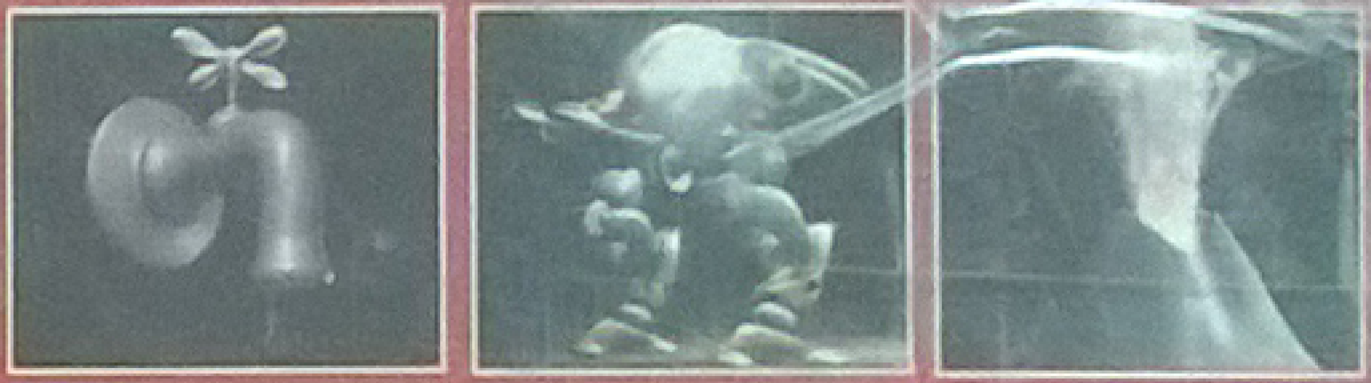
Learning Maya™ 5 | Dynamics

Bonus!
Includes *Exploring Maya 5 Features*
DVD
US $49.99 value

Learning Maya 5 | Dynamics is your solution to creating stunning effects in Maya®.

New for Maya 5!

If your animations don't follow the laws of physics, this book is the tool you've been looking for. You'll approach Maya Dynamics from an introductory level and explore advanced concepts in a comfortable format. You'll work with both Rigid and Soft Body dynamics, particle animation and rendering, emitters, and much more. Helpful step-by-step exercises guide you as you build smoke, dust, fire, and water. Maya's Dynamics engine will allow you to add realism to your animations and create realistic effects without having to go back to school. Learning Maya 5 | Dynamics will help you prove that you don't need to be a rocket scientist to make one fly!



Features:

Each chapter includes a theoretical discussion to help you understand the concepts at play, as well as easy-to-follow step-by-step tutorials so that you can start working right away. This book also includes a DVD-ROM featuring chapter-by-chapter overviews offered by the book's writer, John Patton, an Alias|Wavefront™ Certified Instructor. Gain additional insight and assistance from these valuable bonus discussions.

What you need to use this book:

- Maya Complete™, Maya Unlimited,™ or Maya Personal Learning Edition™ software.
- A DVD-ROM drive to play Chapter Overviews.
- A CD-ROM drive to access Tutorial Support Files.
- QuickTime® 5.0 or higher.

With this book you will:

- Create and animate Rigid Body objects.
- Create and animate particles using fields, goals, ramps and expressions.
- Render particles with hardware graphics and software techniques.
- Dynamically animate NURBS and polygonal surfaces using Soft Bodies.
- Utilize Maya Artisan functionality in conjunction with Maya Dynamics tools.
- Instance geometry with particle motion.
- Use Rigid Body constraints and Soft Body springs.
- Optimize and troubleshoot dynamics scenes.

And much more…

Support Files include:

- Tutorial support files and movies
- Instructor-led chapter overviews

Alias|wavefront®

Visit our websites for more information:
www.aliaswavefront.com and *www.sybex.com*

© 2003 Alias|Wavefront, a division of Silicon Graphics Limited. All rights reserved. The Alias|Wavefront logo is a registered trademark and the Maya logo, Maya Complete, Learning Maya, Maya Techniques, the Maya Press logo, Maya Personal Learning Edition, and Maya Unlimited are trademarks of Alias|Wavefront, a division of Silicon Graphics Limited. Maya is a registered trademark of Silicon Graphics, Inc. in the United States and other countries worldwide exclusively used by Alias|Wavefront, a division of Silicon Graphics Limited. The Sybex logo is a registered trademark of Sybex, Inc. QuickTime is a registered trademark of Apple Computer, Inc. All other trademarks, trade names, service marks and product names mentioned herein are property of their respective owners.

MAYA PRESS™
SYBEX®

US $59.99
CDN $89.95
UK £45.99

7 73736 06053 1

ISBN 1-894893-42-5
9 781894 893428